UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

VEHICLE SCHEDULING PROBLEM WITH MULTIPLE VEHICLE TYPES

by

Jacques A. Ferland*
Philippe Michelon**

*Jacques A. Ferland, Département d'informatique et de recherche opérationnelle et Centre de recherche sur les transports, Université de Montréal.

**Philippe Michelon, Département de mathématiques appliquées, Université de Clermont.

This research was supported by a NSERC grant (A8312).

Cette étude a été publiée grâce à une subvention du fonds F.C.A.R. pour l'aide et le soutien à la recherche.

Université de Montréal

Département d'informatique et
de recherche opérationnelle - Publication #608

Centre de recherche sur les transports - Publication #516

Mars 1987

Dépôt légal - 2e trimestre 1987

ABSTRACT

The vehicle scheduling problem is specified in terms of a set of tasks to be executed with a fleet of multiple vehicle types. The purpose of this paper is to formulate the problem and to show that the heuristic and exact methods developed for the vehicle scheduling problem with time windows and with a single type of vehicle can be extended in a straightforward fashion to the multiple vehicle types problem.

Keywords: Vehicle Scheduling, transportation, network programming.

RESUME

Le problème est défini en termes d'un ensemble de tâches qui doivent être complétées par une flotte de véhicules de différents types. Le problème est d'abord formulé pour permettre d'illustrer qu'il est possible d'adapter simplement les différentes approches heuristiques et exactes utilisées pour traiter ce problème avec un type unique de véhicules.

1. INTRODUCTION

The vehicle scheduling problem can be specified in terms of a set of tasks T to be executed with a fleet of vehicles. Each task i ($1 \leq i \leq n$) has time and space (location) characteristics specified as follows:

- a start location SL_i
- an end location EL_i
- a start time ST_i .

In most realistic contexts, the starting times ST_i are not fixed but rather belong to some given time intervals $[a_i , b_i]$ referred to as "time windows". Hence for each task i, a_i and b_i are the earliest and the latest starting times, respectively. It is also assumed that task i can be executed only by a vehicle having a capacity exceeding q_i .

The fleet of vehicles available can be characterized by its size (the number of vehicles) and by the types of vehicles. In this paper we are concerned with the problem in which several types of vehicles are available. Each type k ($1 \leq k \leq K$) of vehicles is characterized by its capacity h_k and its speed v_k .

To execute task i a vehicle of type k must be at location SL_i at time ST_i . Assume that a vehicle of type k requires D_i^k units of time to execute task i . Then at time $ST_i + D_i^k$, it is located at EL_i and available to execute other tasks. The problem is to partition the set of tasks into sequences. Since all tasks of a sequence are to be executed by the same vehicle, it follows that no two of these can be executed simultaneously and the vehicle type selected for the sequence must be acceptable for all of its tasks. Furthermore task j can follow task i in a sequence executed by a vehicle of type k if, taking into account the time t_{ij}^k required for the vehicle to go from EL_i to SL_j , it is available at time ST_j at SL_j ; i.e.

$$ST_i + D_i^k + t_{ij}^k \leq ST_j$$

where $t_{ij}^k + D_i^k$ depends on the vehicle speed v_k.

The objective function to be minimized can be specified in terms of operating costs, or in terms of investment costs, or in terms of both. For instance, the objective might be to reduce the fleet size to a minimum.

The vehicle scheduling problem is presented and analyzed in Bodin et al.[1] where procedures for handling several variants are summarized. More recently the vehicle scheduling problem with time windows has been analyzed by several authors (Desrosiers et al.[2,3,4,5,6] Graham and Nuttle[7], Orloff[8], Swersey and Ballard[9], Levin[10]). Ferland[11] summarizes the main approaches to handle this problem when a single type of vehicles is used. The purpose of this paper is to indicate how these procedures can be extended to handle the problem with multiple vehicle types.

2. Problem formulation

A mathematical formulation is derived for this problem using the quasi-precedence graphs (N_k, A_k) of the tasks associated with vehicle types k and characterized as follows:

i) With each task i such that $q_i \le h_k$, associate a node $i \in N_k$. Two nodes, s and t, are associated with the depot to indicate where the vehicles start and finish.

ii) An arc (i, j) exists (i.e. $(i, j) \in A_k$, or task j can be executed after task i by a vehicle of type k) if

$$a_i + D_i^k + t_{ij}^k \le b_j .$$

Furthermore, for each task $i \in N_k$, introduce arcs (s,i) and (i,t) in A_k allowing i to be the first or the last task of a sequence, respectively.

In terms of these oriented graphs the problem is to partition the nodes into a set of paths (or sequences), each of these starting at s and ending at t in one of the graphs (N_k, A_k). The time window restrictions are to be satisfied for all tasks, and the objective is to minimize the total cost. The formulation is summarized as follows:

Problem (P) $\quad \text{Min} \sum_{k=1}^{K} \sum_{(i,j)\in A_k} c_{ij}^k x_{ij}^k$

$$\text{Subject to} \sum_{i \in P_j^k} x_{ij}^k = y_j^k \qquad j \in N_k - \{s, t\}, \quad 1 \le k \le K \tag{2.1}$$

$$\sum_{j \in S_i^k} x_{ij}^k = y_i^k \qquad i \in N_k - \{s, t\}, \quad 1 \le k \le K \tag{2.2}$$

$$\sum_{k \in A_i} y_i^k = 1 \qquad 1 \le i \le n \tag{2.3}$$

$$x_{ij}^k > 0 \Rightarrow ST_i + D_i^k + t_{ij}^k \le ST_j \qquad (i, j) \in \bar{A}_k, \ 1 \le k \le K \tag{2.4}$$

$$a_i \le ST_i \le b_i \qquad 1 \le i \le n \tag{2.5}$$

$$x_{ij}^k = 0 \text{ or } 1 \qquad (i, j) \in A_k, \qquad 1 \le k \le K$$

$$y_i^k = 0 \text{ or } 1 \qquad 1 \le i \le n, \qquad 1 \le k \le K$$

where $\Lambda_i = \{k : q_i \le h_k, 1 \le k \le K\}$, $\overline{A}_k = \{(i, j) \in A_k : i \ne s, t ; j \ne s, t\}$. Furthermore for $i \in N_k$, $P_i^k = \{j \in N_k : (j, i) \in A_k\}$ and $S_i^k = \{j \in N_k : (i, j) \in A_k\}$, and the decision variables are defined as follows:

$$x_{ij}^k = \begin{cases} 1 & \text{if a vehicle of type } k \text{ traverses arc } (i, j) \in A_k \\ 0 & \text{otherwise} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{if task } i \text{ is executed with a vehicle of type } k \\ 0 & \text{otherwise .} \end{cases}$$

The linear costs c_{ij}^k are specified according to the objective of the problem. For instance, if the objective is to minimize the investment cost, then

$$c_{ij}^k = \begin{cases} u^k & \text{if } i = s \\ 0 & \text{if } i \ne s \end{cases}$$

where u^k is equal to the investment cost for vehicle type k. If the objective is to minimize investment and running time, then

$$c_{ij}^k = \begin{cases} D_i^k + t_{ij}^k & \text{if } i \text{ and } j \text{ are different from } s \text{ and } t \\ u^k & \text{if } i = s \\ D_i^k & \text{if } j = s \end{cases}$$

where the magnitude of u^k relative to t_{ij}^k and D_i^k measures the importance of investment versus running time.

Constraints (2.1), (2.2) and (2.3) ensure that each task is completed by exactly one vehicle. Constraints (2.4) and (2.5) are the time window constraints.

Several methods have been proposed to solve problem (P) when $K = 1$. Some of these are heuristic methods generating "good" solutions. Other methods generate optimal solutions using "branch-and-bound" procedures. These methods are now extended to handle problem (P) for arbitrary K.

3. Heuristic methods

Three different heuristic methods are now extended. The first method (Levin[10] and by Swersey and Ballard[9]) involves the solution of a discrete approximation of problem (P) specified in Section 2. The second method (Orloff[8]) is a two-phase procedure : phase 1 generates a feasible solution and phase 2 improves on it. The third method (Bodin and Berman[1] and Desrosiers et al.[2]) finds a feasible solution by solving a sequence of transportation problems.

3.1. Discrete approximation

The time window interval of each task is discretized in the sense that specific moments in the time interval are selected as candidate start times for the task. To derive the model, replace each variable ST_i in problem (P) by a set of binary decision variables y_v^k associated with candidate start times for task i. With these variables we specify a mathematical model by referring to precedence graphs $(\tilde{N}_k , \tilde{A}_k)$ of the tasks associated with vehicle types k and characterized as follows:

i) If task i is such that $q_i \le h_k$, then with each of the specific starting times for task i, associate a node $v \in \tilde{N}_k$. Nodes $s, t \in \tilde{N}_k$ are associated with the depot.

ii) An arc (τ, v) exists if τ and v are specific starting times associated with different tasks i and j and if a vehicle of type k can execute task j starting at the specific time v after task i starting at specific time τ. Also, for each $v \in \tilde{N}_k - \{s, t\}$, arcs (s, v) and $(v, t) \in \tilde{A}_k$.

The mathematical model is summarized as follows:

$$\text{Min} \quad \sum_{k=1}^{K} \sum_{(\tau,v)\in \tilde{A}_k} c_{\tau v}^k x_{\tau v}^k$$

$$\text{Subject to} \quad \sum_{\tau \in \tilde{P}_v^k} x_{\tau v}^k = y_v^k \qquad v \in \tilde{N}_k - \{s, t\}, \qquad 1 \le k \le K$$

$$\sum_{v \in \tilde{S}_\tau^k} x_{\tau v}^k = y_\tau^k \qquad \tau \in \tilde{N}_k - \{s, t\}, \qquad 1 \le k \le K$$

$$\sum_{k \in \Lambda_i} \sum_{v \in \Gamma_i^k} y_v^k = 1 \qquad 1 \le i \le m$$

$$x_{\tau v}^k = 0 \text{ ou } 1 \qquad (\tau, v) \in \tilde{A}_k, \qquad 1 \le k \le K$$

$$y_v^k = 0 \text{ ou } 1 \qquad v \in \tilde{N}_k - \{s, t\}, \qquad 1 \le k \le K$$

where $\Gamma_i^k = \{v \in \tilde{N}_k : v \text{ is a specific start time of } i\}$, $\tilde{P}_v^k = \{\tau \in \tilde{N}_k : (\tau, v) \in \tilde{A}_k\}$, and $\tilde{S}_\tau^k = \{v \in \tilde{N}_k : (\tau, v) \in \tilde{A}_k\}$. Furthermore, $c_{\tau v}^k$ is the cost for a vehicle of type k to traverse the arc (τ, v), and the decision variables

$$x_{\tau v}^k = \begin{cases} 1 & \text{if a vehicle of type } k \text{ traverses arc } (\tau, v) \in \tilde{A}_k \\ 0 & \text{otherwise} \end{cases}$$

$$y_v^k = \begin{cases} 1 & \text{if a task with specific starting time } v \text{ is executed with a vehicle of type } k, \\ 0 & \text{otherwise.} \end{cases}$$

Swersey and Ballard[9] have used this model with $K = 1$ in the context of school busing where the objective is to reduce the fleet size. They report that, for the linear programming relaxation of the models (where binary constraints on $x^k_{\tau v}$ and y^k_v are replaced by $0 \le x^k_{\tau v} \le 1$ and $0 \le y^k_v \le 1$) , the optimal solution was integer in many of their examples.

3.2. Matching methods

To extend the two-phase method proposed by Orloff[8] , consider the following oriented graph:

i) With each task i associate a node i .

ii) With each pair of nodes i, j, associate two arcs:

- arc (i, j) with cost

$$c_{ij} = \underset{k \in \Lambda_i \cap \Lambda_j}{\text{Min}} \{t^k_{ij} + D^k_i : \; a_i + D^k_i + t^k_{ij} \le b_j\}$$

- arc (j, i) with cost

$$c_{ji} = \underset{k \in \Lambda_i \cap \Lambda_j}{\text{Min}} \{t^k_{ji} + D^k_j : \; a_j + D^k_j + t^k_{ji} \le b_i\}$$

where the minimum over an empty set is equal to $+\infty$ by convention.

During the first iteration of phase 1, a minimum weighted matching problem is solved on the preceding graph. This is essentially equivalent to identifying a set of arcs such that each node is the extremity of one and only one arc (i.e., no two arcs have a node in common) and the sum of their cost is minimized. If an arc (i, j) belongs to this set, and if $c_{ij} < +\infty$, then collapse nodes i and j into one node i' and specify accordingly a time window

interval $[a_{i'},b_{i'}]$ and the set $\Lambda_{i'} = \Lambda_i \cap \Lambda_j$ for this pair of tasks assumed to be collapsed into one. Then specify a new oriented graph with some of the tasks collapsed into one, and repeat the procedure.

When no further matchings are possible (i.e., when the optimal solution of the minimum weighted matching problem includes only arcs with cost $+\infty$), each node of the current graph corresponds to the schedule of a vehicle executing the sequence of tasks collapsed at this node. This solution is feasible for problem (P) , but is not optimal in general.

During phase 2, the solution is improved through an exchange procedure of tasks between sequences generated in phase 1. It can be viewed as an application of a restricted version of the Lin[12] exchange procedure applied to a travelling salesman tour going through all tasks and generated by linking to one another the sequences generated in phase 1. Furthermore, for each sequence of tasks, the cheapest vehicle is used.

3.3. Assignment method

At each iteration of the procedure, a transportation problem is specified in terms of two subsets of tasks, T_1 , and T_2 . The subset T_1 , is the set of source-tasks including the tasks most recently executed by the vehicles now available for other tasks. The subset T_2 includes a subset of sink-tasks to be executed next.

For the first iteration, the subset T_1 includes the tasks in T that cannot be executed after any other task. Hence $j \in T_1$ if $a_i + \underset{k \in \Lambda_j}{\text{Min}}\{D_i^k + t_{ij}^k\} > b_j$ for all $i \in T$, $i \neq j$. Also, if two tasks can follow each other but cannot be executed after any other, then the task requiring the cheapest vehicle is included in T_1 . If they require the same cheapest type of vehicle, the shortest task is included in T_1 . Hence, each task in T_1 should be the first one of a different

sequence to which the cheapest acceptable vehicle type is assigned.

Now after completing a task in T_1 , a vehicle is available for other tasks. The sequences are built up by solving a series of transportation problems. To determine the subset T_2 of sink-tasks to be executed next, replace T by $T - T_1$, and repeat the procedure used before to define T_1 .

Each node $i \in T_1$ represents the latest task completed in some sequence for which

i) k_i denotes the vehicle type actually assigned,

ii) $\overline{\Lambda}_i$ denotes the set of vehicle types acceptable for all tasks in the sequence,

iii) if the vehicle type $k(k \in \overline{\Lambda}_i)$ is used, then $a_i^k + D_i^k$ denotes the earliest time at which it is available after executing task i .

For each pair of nodes $i \in T_1$, $j \in T_2$, define an assignment cost ac_{ij} for using the vehicle of task i to execute task j as follows

$$ac_{ij} = \begin{cases} t_{ij}^{k_i} + D_j^{k_i} & \text{if } a_i^{k_i} + D_i^{k_i} + t_{ij}^{k_i} \le b_j \\ \underset{k \in \Lambda_i \cap \Lambda_j}{\text{Min}} \{u_k - u_{k_i} : \ a_i^k + D_i^k + t_{ij}^k \le b_j\} & \text{otherwise} \end{cases}$$

Note that if k_i is not acceptable for task j in the sequence, then $\underset{k \in \overline{\Lambda}_i \cap \Lambda_j}{\text{Min}} \{u_k - u_{k_i} : \ a_i^k + D_i^k + t_{ij}^k \le b_j\}$ estimates the increase investment cost for using an alternate acceptable type k . Introduce a dummy source $\overline{s}$ and a dummy sink $\overline{t}$. For all $j \in T_2$ define an assignment cost $ac_{\overline{s}j} = \underset{k \in \Lambda_j}{\text{Min}}\{u_k\}$ to indicate that a new sequence is initiated with task j using the cheapest acceptable vehicle. Finally, let $ac_{i\overline{t}} = 0$.

To complete the specification of the transportation problem, assume that one unit (vehicle) is available at each node in T_1 , one unit is required at each node in T_2 , $|T_2|$ units are available at $\bar{s}$ and $|T_1|$ units are required at $\bar{t}$. All arcs have a unit capacity except $(\bar{s}, \bar{t})$ for which capacity is equal to $\text{Max}\{|T_1| , |T_2|\}$.

This transportation problem is solved. If $i \in T_1$ is assigned to $\bar{t}$, then i remains in T_1 . If $\bar{s}$ is assigned to $j \in T_2$, then a new sequence is initiated, and j is introduced into T_1. If $i \in T_1$ is assigned to $j \in T_2$, then the vehicle executing i is also used for j . Then i is eliminated from T_1 , and j is introduced into T_1 to replace it.

The procedure is repeated. The current set T_1 of source-tasks is used. The set T_2 is determined using the procedure described before. The process stops when all tasks are included in some sequence. This method yields a feasible solution which is not optimal in general.

4. Exact Methods

Two branch-and-bound exact methods have been proposed to find optimal solutions to problem (P) when $K = 1$. The first method presented by Desrosiers et al.[5] uses a column generation technique to solve a set partitioning problem equivalent to problem (P) . In the second method, at each node of the branch-and-bound tree we have to solve a problem of type (P) where time window constraints are relaxed. This is presented in Desrosiers et al.[4] .

The extension of these methods to the case $K > 1$ is straightfoward. For the sake of completeness, we summarize the main steps of these extended methods.

4.1. Column generation technique

Let Ω denote a set of task sequences which satisfy the time window constraints. An acceptable vehicle is associated with each of these sequences. Use the decision variables z_r , $r \in \Omega$ to specify the following set partitioning problem

Problem (*SP*) Min $\sum_{r \in \Omega} c_r z_r$

Subject to $\sum_{r \in \Omega} \delta_i^r z_r = 1 \qquad 1 \le i \le n$

$z_r = 0 \text{ or } 1 \qquad r \in \Omega$

where

$$z_r = \begin{cases} 1 & \text{if sequence } r \in \Omega \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_i^r = \begin{cases} 1 & \text{if task } i \text{ belongs to sequence } r \\ 0 & \text{otherwise} \end{cases}$$

To determine the cost c_r , for all $i, j \in T$, denote

$$\delta_{ij}^r = \begin{cases} 1 & \text{if task } j \text{ follows task } i \text{ in the } r \text{ th sequence} \\ 0 & \text{otherwise} , \end{cases}$$

and for all $i \in T$,

$$\delta_{si}^r = \begin{cases} 1 & \text{if task } i \text{ is the first of the } r \text{ th sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{ti}^r = \begin{cases} 1 & \text{if task } i \text{ is the last of the } r \text{ th sequence} \\ 0 & \text{otherwise} . \end{cases}$$

Hence, assuming that a vehicle of type k is executing the r th sequence, it follows that

$$c_r = \sum_{(i,j)\in A_k} c_{ij}^k \delta_{ij}^r .$$

At each iteration of the procedure the linear programming relaxation of a problem (SP) associated with a given Ω is solved. Denote by σ_i the optimal multiplier associated with the constraint $\sum_{r\in\Omega} \delta_i^r z_r = 1$. Pricing out the r th sequence, we obtain

$$\sum_{(i,j)\in A_k} c_{ij}^k \delta_{ij}^r - \sum_{i=1}^{n} \delta_i^r \sigma_i .$$

The approach is to generate new sequences not already in Ω which might reduce the overall cost of the set partitioning problem. These new sequences have negative relative costs when evaluated with the optimal multipliers σ_i . It is easy to see that such sequences are generated by solving a shortest path problem with the additional time window constraints in each of the oriented graphs (N_k, A_k) where the cost of arc (i, j) is set equal to $(c_{ij}^k - \sigma_i)$. Desrosiers et al.[13] propose an algorithm to solve this problem.

Desrosiers et al.[5] method is extended to the case when K is arbitrary by embedding this procedure into a branch-and-bound method to obtain an optimal integer solution to a set partitioning problem (SP) equivalent to (P) .

4.2. Time window constraints relaxation

Consider the second branch-and-bound procedure proposed by Desrosiers et *al.*[4] . At each node of the tree, a problem of type (P) must be solved where some variables are fixed and where the time window constraints are relaxed. The branching is specified to eliminate subsequences where time window constraints are not satisfied or including a cycle of tasks (i.e.

the last task of the subsequence is linked to the first).

The approach is valid for arbitrary values of K. When $K = 1$, if the time window constraints are relaxed, then (P) reduces to a simple network flow problem. But when K is arbitrary, this problem is more complicated.

5. Relative efficiency of the methods

Since the methods for handling problem (P) when $K > 1$ are straightforward extensions of those for $K = 1$, their relative efficiency should be preserved. Hence the efficiency of the column generation approach reported by Desrosiers et al.[5] should justify its use whenever problem size does not become prohibitive. On the other hand heuristic procedures can handle larger problems in general and generate "good" solutions in many cases.

References

1. L. BODIN, A. ASSAD, M. BALL and B. GOLDEN (1983) The state of the art in routing and scheduling of vehicles and crews. Computers and Operations Research 10, 69-211.

2. J. DESROSIERS, J.A. FERLAND, J.-M. ROUSSEAU, G. LAPALME and L. CHAPLEAU (1986) TRANSCOL: A multi-period school bus routing and scheduling system. In TIMS Studies in Management Sciences 22 (E.J. IGNALL, A.J. SWERSEY, Eds.), pp. 47-71, North-Holland, Amsterdam.

3. J. DESROSIERS, F. SOUMIS, M. SAUVE (1987) Lagrangean methods for solving the minimal M-TSP with time windows. Management Science (forthcoming)

4. J. DESROSIERS, F. SOUMIS, M. DESROCHERS and M. SAUVE (1985) Routing and scheduling with time windows solved by network relaxation and branch-and-bound on time variables. In Computer Sceduling of Public Transport II (J.-M. ROUSSEAU, Ed.) pp. 451-471 . North-Holland, Amsterdam.

5. J. DESROSIERS, F. SOUMIS and M. DESROCHERS (1984) Routing with time windows by column generation. Networks 14, 545-565.

6. J. DESROSIERS, F. SOUMIS and M. SAUVE (1984) Lagrangean relaxation for routing with time windows. In Operational Research (J. P. BRANS. Ed.) pp. 673-684. North-Holland, Amsterdam.

7. D. GRAHAM and L.W. NUTTLE (1984) A comparison of heuristics for a school bus scheduling problem. I E technical report No 84-3, Department of Industrial Engineering, North Carolina State University.

8. C. ORLOFF (1976) Route constrained fleet scheduling. Transportation Science 10, 149-168.

9. A.J. SWERSEY and W. BALLARD (1984) Scheduling school buses. Management Science 30, 844-853.

10. A. LEVIN (1971) Scheduling and fleet routing models for transportation systems. Transportation Science 5, 232-255.

11. J.A. FERLAND (1985) The single depot vehicle scheduling problem. Publication #385, Centre de recherche sur les transports, Université de Montréal.

12. S. LIN (1965) Computer solutions of the travelling salesman problem. Bell System Technical Journal 44, 2245-2269.

13. J. DESROSIERS, P. PELLETIER, F. SOUMIS (1983) Plus court chemin avec contraintes d'horaire. RAIRO 17, 357-377.